

Bayesian Model Selection Methods for Mutual and Symmetric k -Nearest Neighbor Classification

Hyun-Chul Kim

Abstract—The k -nearest neighbor classification method (k -NNC) is one of the simplest nonparametric classification methods. The mutual k -NN classification method (M k NNC) is a variant of k -NNC based on mutual neighborhood. We propose another variant of k -NNC, the symmetric k -NN classification method (S k NNC) based on both mutual neighborhood and one-sided neighborhood. The performance of M k NNC and S k NNC depends on the parameter k as the one of k -NNC does. We propose the ways how M k NN and S k NN classification can be performed based on Bayesian mutual and symmetric k -NN regression methods with the selection schemes for the parameter k . Bayesian mutual and symmetric k -NN regression methods are based on Gaussian process models, and it turns out that they can do M k NN and S k NN classification with new encodings of target values (class labels). The simulation results show that the proposed methods are better than or comparable to k -NNC, M k NNC and S k NNC with the parameter k selected by the leave-one-out cross validation method not only for an artificial data set but also for real world data sets.

Index Terms— k -NN classification, mutual k -NN classification, symmetric k -NN classification, Selecting k in k -NN, symmetric k -NN regression, Bayesian symmetric k -NN regression, Gaussian processes, Bayesian model selection

I. INTRODUCTION

One of the well-known nonparametric classification methods is k -nearest neighbor (k -NN) classification method [1], [2], [3]. It uses one of the simplest rules among nonparametric classification methods. It assigns to a given test data point the most frequent class label appearing in the set of k nearest data points to the test data point. Performance of k -NN classifiers is influenced by the distance measure [4] and the parameter k ¹ [5], [6]. So it is an important issue to select the best distance measure and the best parameter k in k -NN classification. In this paper we focus on the selection of the best parameter k in the variates of k -NN classification although the selection of the best distance measure is also important.

[5] proposed an approximate Bayesian approach to k -NN classification, where a single parameter k was not selected but its posterior distribution was estimated. It was not exactly probabilistic because of the missing of the proper normalization constant in the model as mentioned in [7]. It provided the class probability for a test data point and the approximate distribution of the parameter k by MCMC methods. It was followed by an alternative model with likelihood-based inference and a method to select the best parameter k based

on BIC (Bayesian information criterion) method [8]. While those models are not fully probabilistic, [7] proposed a full Bayesian probabilistic model for k -NN classification based on a symmetrized Boltzmann modelling with various kinds of sampling methods including a perfect sampling. Due to the symmetrized modification, their model does not fully reflect k -NN classification any more (e.g. it does not have asymmetry such as the one in k -NN classification.). So the most probable k in their model may not be optimal in k -NN classification.

[6] has proposed another method to select the optimal parameter k based on approximate Bayes risk. They modelled class probabilities for each training data point based on k -NN density estimation in the leave-one-out manner. Starting from those class probabilities by applying Bayes' rule they get the accuracy index $\alpha(k)$. They proved that $1 - \alpha(k)$ asymptotically converges to the optimal Bayes risk. Their simulation results showed that their proposed methods were better than cross-validation and likelihood cross-validation techniques.

Mutual k -NN (M k NN) classification is a variate of k -NN classification based on mutual neighborhood rather than one-sided neighborhood. M k NN concept was applied to clustering tasks [9], [10]. More recently, M k NN methods have been applied to classification [11], outlier detection [12], object retrieval [13], clustering of interval-valued symbolic patterns [14], and regression [15]. [16] used M k NN concept to semi-supervised classification of natural language data and showed that the case of using M k NN concept consistently outperform the case of using k -NN concept.

We propose another variate of k -NN classification, symmetric k -NN (S k NN) classification motivated by a symmetrized modelling used in [7]. S k NN consider neighbors both with mutual neighborhood and one-sided neighborhood. In S k NN classification, one-sided neighbors contribute to the decision in the same way as in k -NN classification, and mutual k -nearest neighbors contribute to the decision twice more than one-sided k nearest neighbors.

We propose Bayesian methods to select the parameter k for M k NN² and S k NN classification. This paper does not propose Bayesian probabilistic models for M k NN and S k NN classification, but model selection methods for them in the Bayesian evidence framework are proposed. The methods are based on Bayesian M k NN and S k NN regression methods, with which M k NN and S k NN classification can be done. A model selection method for S k NN classification is related to the estimation of the parameter k in [7], because the model

H.-C. Kim is with R² Research, Seoul, South Korea.
E-mail: hckim.sr@gmail.com

¹More accurately k should be called the hyperparameter since k -NN is a nonparametric method, but in this paper we also call it the parameter according to the convention.

²To our knowledge no Bayesian model selection method for M k NN classification has been proposed.

proposed in [7] can be regarded a Bayesian probabilistic model for $SkNN$ classification.

The paper is organized as follows. In section II we describe mutual k -NN and symmetric k -NN regression, and their Bayesian extensions with the selection method for the parameter k . In Section III we explain how we can do $MkNN$ and $SkNN$ classification with Bayesian $MkNN$ and $SkNN$ regression methods with their own selection schemes for the parameter k . In section IV we show simulation results for an artificial data set and real-word data sets. Finally a conclusion is drawn.

II. BAYESIAN MUTUAL AND SYMMETRIC k -NEAREST NEIGHBOR REGRESSION

A. Mutual and Symmetric k -Nearest Neighbor Regression

Let $\mathcal{N}_k(\mathbf{x})$ be the set of the k nearest neighbors of \mathbf{x} in \mathcal{D}_n , $\mathcal{N}'_k(\mathbf{x}_i)$ the set of k nearest neighbors of \mathbf{x}_i in $(\mathcal{D}_n \setminus \{\mathbf{x}_i\}) \cup \{\mathbf{x}\}$. The set of mutual nearest neighbors of \mathbf{x} is defined as

$$\mathcal{M}_k(\mathbf{x}) = \{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}) : \mathbf{x} \in \mathcal{N}'_k(\mathbf{x}_i)\}. \quad (1)$$

Then, the mutual k -nearest neighbor regression estimate is defined as

$$m_n^{MkNNR}(\mathbf{x}) = \begin{cases} \frac{1}{M_k(\mathbf{x})} \sum_{i:\mathbf{x}_i \in \mathcal{M}_k(\mathbf{x})} y_i & \text{if } M_k(\mathbf{x}) \neq 0; \\ 0 & \text{if } M_k(\mathbf{x}) = 0. \end{cases} \quad (2)$$

where $M_k(\mathbf{x}) = |\mathcal{M}_k(\mathbf{x})|$ [15].

Motivated by the symmetrised modelling for the k -NN classification in [7], we define the symmetric k -nearest neighbor regression estimate as

$$\begin{aligned} m_n^{SkNNR}(\mathbf{x}) &= \frac{1}{N_k(\mathbf{x}) + N'_k(\mathbf{x})} \sum_{i:\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} (\delta_{\mathbf{x}_i \in \mathcal{N}_k(\mathbf{x})} + \delta_{\mathbf{x} \in \mathcal{N}'_k(\mathbf{x}_i)}) y_i. \end{aligned} \quad (3)$$

where $N_k(\mathbf{x}) = |\mathcal{N}_k(\mathbf{x})|$ and $N'_k(\mathbf{x}) = |\mathcal{N}'_k(\mathbf{x})|$.

B. Bayesian Mutual and Symmetric k -NN Regression via Gaussian Processes

1) *Gaussian Process Regression:* Assume that we have a data set D of data points \mathbf{x}_i with continuous target values y_i : $D = \{(\mathbf{x}_i, y_i) | i = 1, 2, \dots, n\}$, $X = \{\mathbf{x}_i | i = 1, 2, \dots, n\}$, $\mathbf{y} = [y_1, y_2, \dots, y_n]^T$. We assume that the observations of target values are noisy, and set $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $f(\cdot)$ is a target function to be estimated and $\epsilon_i \sim \mathcal{N}(0, v_1)$. A function $f(\cdot)$ to be estimated given D is assumed to have Gaussian process prior, which means that any collection of functional values are assumed to be multivariate Gaussian.

The prior for the function values $\mathbf{f} = [f(\mathbf{x}_1) f(\mathbf{x}_2) \dots f(\mathbf{x}_n)]^T$ is assumed to be Gaussian:

$$p(\mathbf{f} | X, \Theta_f) = \mathcal{N}(\mathbf{0}, \mathbf{C}_f). \quad (4)$$

Then the density function for the target values can be described as follows.

$$p(\mathbf{y} | X, \Theta) = \mathcal{N}(\mathbf{0}, \mathbf{C}_f + v_1 \mathbf{I}) \quad (5)$$

$$= \mathcal{N}(\mathbf{0}, \mathbf{C}), \quad (6)$$

where \mathbf{C} is a matrix whose elements C_{ij} is a covariance function value $c(\mathbf{x}_i, \mathbf{x}_j)$ of \mathbf{x}_i , \mathbf{x}_j and Θ is the set of hyperparameters in the covariance function.

It can be shown that GPR provides the following distribution of target value $f_{\text{new}} (= f(\mathbf{x}_{\text{new}}))$ given a test data \mathbf{x}_{new} :

$$p(f_{\text{new}} | \mathbf{x}_{\text{new}}, D, \Theta) = \mathcal{N}(\mathbf{k}^T \mathbf{C}^{-1} \mathbf{f}, \kappa - \mathbf{k}^T \mathbf{C}^{-1} \mathbf{k}), \quad (7)$$

where $\mathbf{k} = [c(\mathbf{x}_{\text{new}}, \mathbf{x}_1) \dots c(\mathbf{x}_{\text{new}}, \mathbf{x}_n)]^T$, $\kappa = c(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{new}})$. The variance of the target value f_{new} is related to the degree of its uncertainty. We can select the proper Θ by maximizing the marginal likelihood $p(\mathbf{y} | X, \Theta)$ [17], [18], [19], or we can average over the hyperparameters with MCMC methods [17], [20].

2) *Laplacian-based Covariance Matrix:* The combinatorial Laplacian \mathbf{L} is defined as follows.

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \quad (8)$$

where \mathbf{W} is an $N \times N$ edge-weight matrix with the edge weight between two points $\mathbf{x}_i, \mathbf{x}_j$ given as $w_{ij} (= w(\mathbf{x}_i, \mathbf{x}_j))$ and $\mathbf{D} = \text{diag}(d_1, \dots, d_N)$ is a diagonal matrix with diagonal entries $d_i = \sum_j w_{ij}$.

Similarly to [21], to avoid the singularity we set Laplacian-based covariance matrix as

$$\mathbf{C} = (\mathbf{L} + \sigma^2 \mathbf{I})^{-1} = \tilde{\mathbf{C}}^{-1}. \quad (9)$$

Then, we have Gaussian process prior as follows.

$$p(\mathbf{y} | X, \Theta) = \mathcal{N}(\mathbf{0}, \mathbf{C}), \quad (10)$$

The predictive distribution for y_{new} is as follows (See [22] for the detailed derivation).

$$p(y_{\text{new}} | \mathbf{y}, X, \mathbf{x}_{\text{new}}, \Theta) \propto \mathcal{N}\left(-\frac{1}{\tilde{\kappa}} \tilde{\mathbf{k}}^T \mathbf{y}, \frac{1}{\tilde{\kappa}}\right), \quad (11)$$

where

$$\tilde{\kappa} = \sum_{i=1}^N w(\mathbf{x}_{\text{new}}, \mathbf{x}_i) + \sigma^2, \quad (12)$$

$$\tilde{\mathbf{k}}^T = -[w(\mathbf{x}_{\text{new}}, \mathbf{x}_1), w(\mathbf{x}_{\text{new}}, \mathbf{x}_2), \dots, w(\mathbf{x}_{\text{new}}, \mathbf{x}_N)]. \quad (13)$$

The mean and variance of y_{new} is represented as

$$\mu_{y_{\text{new}}} = -\frac{1}{\tilde{\kappa}} \tilde{\mathbf{k}}^T \mathbf{y}_L = \frac{\sum_{i=1}^N w(\mathbf{x}_{\text{new}}, \mathbf{x}_i) y_i}{\sum_{i=1}^N w(\mathbf{x}_{\text{new}}, \mathbf{x}_i) + \sigma^2}, \quad (14)$$

$$\sigma_{y_{\text{new}}}^2 = \frac{1}{\tilde{\kappa}} = \frac{1}{\sum_{i=1}^N w(\mathbf{x}_{\text{new}}, \mathbf{x}_i) + \sigma^2}. \quad (15)$$

3) *Bayesian Mutual and Symmetric k -NN Regression*: First, we describe Bayesian mutual k -NN regression proposed in [22]. When we replace $w_{ij}(=w(\mathbf{x}_i, \mathbf{x}_j))$ with the function

$$w_{\text{MkNN}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_0 \delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j}, \quad (16)$$

where the relation \sim_k is defined as

$$\mathbf{x}_i \sim_k \mathbf{x}_j = \begin{cases} T & \text{if } j \neq i \text{ and } \mathbf{x}_j \text{ is a } k\text{-nearest neighbor of } \mathbf{x}_i; \\ F & \text{otherwise,} \end{cases} \quad (17)$$

and apply Eq (15), we get Bayesian mutual k -NN regression estimate given a new data \mathbf{x}_{new} as follows.

$$m_n^{\text{BMkNNR}}(\mathbf{x}_{\text{new}}) = \mu_{f_{\text{new}}, \text{MkNN}}, \quad (18)$$

where

$$\begin{aligned} \mu_{f_{\text{new}}, \text{MkNN}} &= \frac{\sum_{i=1}^N w_{\text{MkNN}}(\mathbf{x}_{\text{new}}, \mathbf{x}_i) y_i}{\sum_{i=1}^N w_{\text{MkNN}}(\mathbf{x}_{\text{new}}, \mathbf{x}_i) + \sigma^2} \\ &= \frac{\sum_{i=1}^N \delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j} y_i}{\sum_{i=1}^N \delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j} + \sigma^2 / \sigma_0}. \end{aligned} \quad (19)$$

We have two following theorems about the validity of the covariance matrix with $w_{\text{MkNN}}(\mathbf{x}_i, \mathbf{x}_j)$ and asymptotic property of the regression estimate. (See [22] for the proofs.)

Theorem 1. *Covairance matrix $\tilde{\mathbf{C}}$ with $w_{ij}(=w_{\text{MkNN}}(\mathbf{x}_i, \mathbf{x}_j))$ is valid for Gaussian processes if $\sigma^2 > 0$.*

Theorem 2. *$\mu_{f_{\text{new}}, \text{MkNN}}(=-\frac{1}{\tilde{\kappa}} \tilde{\mathbf{k}}^T \mathbf{y})$ converges to mutual k -NN regression as σ^2 / σ_0 approaches 0.*

Now related to symmetric k -NN regression, we propose Bayesian symmetric k -NN regression.

$$w_{\text{SkNN}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_0 (\delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j}) \quad (20)$$

Similarly to Eq (14) Bayesian symmetric k -NN regression is obtained as follows.

$$m_n^{\text{BMkNNR}}(\mathbf{x}_{\text{new}}) = \mu_{f_{\text{new}}, \text{MkNN}}, \quad (21)$$

where

$$\mu_{f_{\text{new}}, \text{SkNN}} = \frac{\sum_{i=1}^N w_{\text{SkNN}}(\mathbf{x}, \mathbf{x}_i) y_i}{\sum_{i=1}^N w_{\text{SkNN}}(\mathbf{x}, \mathbf{x}_i) + \sigma^2} \quad (22)$$

$$= \frac{\sum_{i=1}^N (\delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j}) y_i}{\sum_{i=1}^N (\delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j}) + \sigma^2 / \sigma_0}. \quad (23)$$

The symmetric k -NN regression estimate in Eq (3) can be described as follows.

$$m_n^{\text{SkNNR}}(\mathbf{x}) = \frac{\sum_{i=1}^N w_{\text{SkNN}}(\mathbf{x}, \mathbf{x}_i) y_i}{\sum_{i=1}^N w_{\text{SkNN}}(\mathbf{x}, \mathbf{x}_i)} \quad (24)$$

We have two following theorems about the validity of the covariance matrix with $w_{\text{SkNN}}(\mathbf{x}_i, \mathbf{x}_j)$ and asymptotic property of the regression estimate. (See Appendix A and B for the proofs.)

Theorem 3. *Covairance matrix $\tilde{\mathbf{C}}$ is valid for Gaussian processes if $\sigma^2 > 0$.*

Theorem 4. *$\mu_{f_{\text{new}}, \text{SkNN}}(=-\frac{1}{\tilde{\kappa}} \tilde{\mathbf{k}}^T \mathbf{y})$ converges to symmetric k -NN regression as σ^2 / σ_0 approaches 0.*

4) *Hyperparameter Selection*: We describe the hyperparameter selection method for Bayesian MkNN regression proposed in [22]. It can be also used for the hyperparameter selection for Bayesian SkNN regression proposed in this paper. We have the set of hyperparameters is $\Theta = \{k, \sigma_0, \sigma\}$, where k is a interger greater than 0. These sets of hyperparameters can be selected through the Bayesian evidence framework by maximizing the log of the marginal likelihood [18] as follows.

$$\Theta^* = \text{argmax}_{\Theta} \mathcal{L}(\Theta), \quad (25)$$

where

$$\mathcal{L}(\Theta) = \log p(\mathbf{y}|\Theta) \quad (26)$$

$$= \log \{ |2\pi \mathbf{C}|^{-\frac{1}{2}} \exp(-\frac{1}{2} \mathbf{y}^T \mathbf{C}^{-1} \mathbf{y}) \} \quad (27)$$

$$= \frac{1}{2} \log |\tilde{\mathbf{C}}| - \frac{1}{2} \mathbf{y}^T \tilde{\mathbf{C}} \mathbf{y} - \frac{N}{2} \log 2\pi, \quad (28)$$

where $\tilde{\mathbf{C}} = \mathbf{L} + \sigma^2 \mathbf{I}$.

For the continuous hyperparameters (e.g., σ, σ_0), the derivative can be used to optimize \mathcal{L} with respect to Θ , where the derivative of \mathcal{L} with respect to θ is given by

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{2} \text{trace}(\tilde{\mathbf{C}}^{-1} \frac{\partial \tilde{\mathbf{C}}}{\partial \theta}) - \frac{1}{2} \mathbf{y}^T \frac{\partial \tilde{\mathbf{C}}}{\partial \theta} \mathbf{y}, \quad (29)$$

where $\tilde{\mathbf{C}} = \mathbf{L} + \sigma^2 \mathbf{I}$. The discrete hyperparameter k can be selected based on the value of \mathcal{L} as

$$K^* = \text{argmax}_k \mathcal{L}(\{k, \sigma, \sigma_0\}). \quad (30)$$

On the other hand, the posterior distributions of the hyperparameters given the data can be inferred by the Bayesian method via Markov Chain Monte Carlo methods similarly to [20], [17]. And the regression estimate can be averaged over the hyperparameters rather than obtained by one fixed set of hyperparameters. This would produce better results but cost more computational power. This approach has not been taken in this paper

III. MUTUAL AND SYMMETRIC k -NN CLASSIFICATION AND BAYESIAN SELECTION METHODS FOR k

A. Mutual and Symmetric k -Nearest Neighbor Classification

Let us assume we have the data set $\mathcal{D}_n = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i \in \mathbf{R}^d$ and $y_i \in \{C_1, C_2, \dots, C_J\}$. We describe mutual and symmetric k -NN classification methods with the notations $\mathcal{N}_k(\mathbf{x})$, $\mathcal{N}'_k(\mathbf{x}_j)$, and $\mathcal{M}_k(\mathbf{x})$ used to describe mutual and symmetric k -nearest neighbor regression in Section II-A. The mutual k -NN classification method is described as

$$m_n^{\text{MkNNC}}(\mathbf{x}) = C_{\text{argmax}_c \{|\{\mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}) | y_j = c\}|\}}. \quad (31)$$

Motivated by the symmetrised modelling used in [7], we describe the symmetric k -NN classification method as

$$m_n^{\text{SkNNC}}(\mathbf{x}) = C_{\text{argmax}_c [|\{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}) | y_j = c\}| + |\{\mathbf{x}_j \in \mathcal{N}'_k(\mathbf{x}_j) | y_j = c\}|]}. \quad (32)$$

It is trivial to show that the class label that the model in [7] estimates with the highest class probability is the same as the one that the above method presents. The model proposed by [7] can be regarded as a full Bayesian model for the symmetric k -NN classification mentioned above.

B. Bayesian Selection Methods for k

In Section II-B we described Bayesian mutual and symmetric k -NN regression methods with the selection schemes for the hyperparameters including k . We show that mutual and symmetric k -NN classification can be done with Bayesian mutual and symmetric k -NN regression methods, if the target values of the data set is encoded properly from class labels. We describe how it can be done for the cases of binary-class and multi-class (more than 2 classes) classification.

1) *Binary-class Classification:* In case of the binary-class classification, we set a new training data set $\mathcal{D}_n^{\text{CR}} = \{(\mathbf{x}_1, y_1^{\text{NE}}), \dots, (\mathbf{x}_n, y_n^{\text{NE}})\}$ with new class label encodings, where

$$y_i^{\text{NE}} = \begin{cases} -1 & \text{if } y_i = C_1 \\ 1 & \text{if } y_i = C_2. \end{cases} \quad (33)$$

Now given a new test data point \mathbf{x} we apply Bayesian $MkNN$ regression for the new training data set $\mathcal{D}_n^{\text{CR}}$, and then we have $MkNN$ classification method based on the result of Bayesian $MkNN$ regression:

$$y_{\text{new}}^{\text{MkNN,NE}} = \text{sgn}(\mu_{f_{\text{new},MkNN}}) \quad (34)$$

$$= \text{sgn}\left(\sum_{i=1}^N \delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}} y_i^{\text{NE}}\right) \quad (35)$$

$$= \text{sgn}(-|\{\mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}_{\text{new}}) | y_j = C_1\}| + |\{\mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}_{\text{new}}) | y_j = C_2\}|). \quad (36)$$

It is also trivial to show that

$$y_{\text{new}}^{\text{MkNN}} = C_{(y_{\text{new}}^{\text{MkNN,NE}}+3)/2} = m_n^{\text{MkNNC}}(\mathbf{x}). \quad (37)$$

For the symmetric k -NN classification, we apply Bayesian $SkNN$ regression for the new training data set $\mathcal{D}_n^{\text{CR}}$, and then we have $SkNN$ classification method based on the result of Bayesian $SkNN$ regression:

$$y_{\text{new}}^{\text{SkNN,NE}} = \text{sgn}(\mu_{f_{\text{new},SkNN}}) \quad (38)$$

$$= \text{sgn}\left(\sum_{i=1}^N (\delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}}) y_i^{\text{NE}}\right) \quad (39)$$

$$= \text{sgn}[-|\{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_{\text{new}}) | y_j = C_1\}| + |\{\mathbf{x}_{\text{new}} \in \mathcal{N}'_k(\mathbf{x}_j) | y_j = C_1\}| + |\{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_{\text{new}}) | y_j = C_2\}| + |\{\mathbf{x}_{\text{new}} \in \mathcal{N}'_k(\mathbf{x}_j) | y_j = C_2\}|]. \quad (40)$$

It is also trivial to show that

$$y_{\text{new}}^{\text{SkNN}} = C_{(y_{\text{new}}^{\text{SkNN,NE}}+3)/2} = m_n^{\text{SkNNC}}(\mathbf{x}). \quad (41)$$

The hyperparameters including k can be selected by the methods described in Section II-B4.

2) *Multi-class Classification:* For the multi-class classification (with more than 2 classes), we present two kinds of methods. First, we use the traditional formulation (*formulation I*) used in multi-class Gaussian process classification [23]. We consider Bayesian mutual and symmetric k -NN regression with J outputs when we have J classes. The outputs are

expressed as f^1, f^2, \dots, f^J . We assume that the $Jn \times Jn$ covariance matrix of the prior of \mathbf{f} is

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{f^1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{f^1} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{f^J} \end{bmatrix}, \quad (42)$$

with the covariance function $\text{Cov}(f_i^j, f_k^l) = \delta(j, l)c(\mathbf{x}_i, \mathbf{x}_k)$. Then we have

$$\mathbf{C} = (\mathbf{D} - \mathbf{W} + \sigma^2 \mathbf{I}_{Jn \times Jn})^{-1} \quad (43)$$

$$\mathbf{C}_{f^l} = (\mathbf{D}_{f^l} - \mathbf{W}_{f^l} + \sigma^2 \mathbf{I}_{n \times n})^{-1} \quad (44)$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{f^1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{f^1} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{W}_{f^J} \end{bmatrix}, \quad (45)$$

where $[\mathbf{W}_{f^l}]_{ij} = w_{MkNN}(\mathbf{x}_i, \mathbf{x}_j)$ for $MkNN$ case, or $w_{SkNN}(\mathbf{x}_i, \mathbf{x}_j)$ for $SkNN$ case. When we set a new encoding for a target value as

$$y_{il}^{\text{NE}_2} = \begin{cases} 1 & \text{if } y_i = C_l; \\ 0 & \text{otherwise,} \end{cases} \quad (46)$$

given a new test data point \mathbf{x} we have the predictive mean

$$\mu_{f_{\text{new},MkNN}}^l = \frac{\sum_{i=1}^N w_{MkNN}(\mathbf{x}_{\text{new}}, \mathbf{x}_i) y_{il}^{\text{NE}_2}}{\sum_{i=1}^N w_{MkNN}(\mathbf{x}_{\text{new}}, \mathbf{x}_i) + \sigma^2} \quad (47)$$

$$= \frac{\sum_{i=1}^N \delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}} y_{il}^{\text{NE}_2}}{\sum_{i=1}^N \delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}} + \sigma^2 / \sigma_0}. \quad (48)$$

Then we have the classification method based on the result of multivariate Bayesian $MkNN$ regression

$$y_{\text{new}}^{\text{MkNN,MUL-I}} = C_{\text{argmax}_l \mu_{f_{\text{new},MkNN}}^l} \quad (49)$$

$$= C_{\text{argmax}_l \sum_{i=1}^N \delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} \cdot \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}} y_{il}^{\text{NE}_2}} \quad (50)$$

$$= C_{\text{argmax}_c |\{\mathbf{x}_j \in \mathcal{M}_k(\mathbf{x}_{\text{new}}) | y_j = c\}|} \quad (51)$$

$$= m_n^{\text{MkNNC}}(\mathbf{x}). \quad (52)$$

Similarly, for symmetric k -NN classification we have the classification method based on the result of multivariate Bayesian $SkNN$ regression

$$y_{\text{new}}^{\text{SkNN,MUL-I}} = C_{\text{argmax}_l \mu_{f_{\text{new},SkNN}}^l} \quad (53)$$

$$= C_{\text{argmax}_l \sum_{i=1}^N (\delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}}) y_{il}^{\text{NE}_2}} \quad (54)$$

$$= C_{\text{argmax}_c [|\{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_{\text{new}}) | y_j = c\}| + |\{\mathbf{x}_{\text{new}} \in \mathcal{N}'_k(\mathbf{x}_j) | y_j = c\}|]} \quad (55)$$

$$= m_n^{\text{SkNNC}}(\mathbf{x}), \quad (56)$$

where

$$\begin{aligned} \mu_{f_{\text{new},SkNN}}^l &= \frac{\sum_{i=1}^N w_{SkNN}(\mathbf{x}_{\text{new}}, \mathbf{x}_i) y_{il}^{\text{NE}_2}}{\sum_{i=1}^N w_{SkNN}(\mathbf{x}_{\text{new}}, \mathbf{x}_i) + \sigma^2} \\ &= \frac{\sum_{i=1}^N (\delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}}) y_{il}^{\text{NE}_2}}{\sum_{i=1}^N (\delta_{\mathbf{x}_{\text{new}} \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_{\text{new}}}) + \sigma^2 / \sigma_0}. \end{aligned} \quad (57)$$

As in [24] we use another formulation (*formulation II*) to avoid a redundancy in the traditional formulation pointed by [20]. We use $J - 1$ outputs only without redundancy,

which are $(J - 1)$ differences among $\{f^1, f^2, \dots, f^J\}$. We define $g_i^{y_i, j} (= f_i^{y_i} - f_i^j)$ for $j \neq y_i$. We set \mathbf{g}_i to $[g_i^{y_i, 1}, \dots, g_i^{y_i, y_i-1}, g_i^{y_i, y_i+1}, \dots, g_i^{y_i, J}]^T$, and set \mathbf{g} to $[\mathbf{g}_1^T, \mathbf{g}_2^T, \dots, \mathbf{g}_n^T]^T$. For \mathbf{g} we have the $(J - 1)n \times (J - 1)n$ covariance matrix \mathbf{C}^{MUL} with the covariance function $\text{Cov}(g_i^{y_i, j}, g_k^{y_k, l}) = (\delta(y_i, y_k) - \delta(y_i, l) - \delta(y_k, j) + \delta(j, l))c(\mathbf{x}_i, \mathbf{x}_k)$ for $y_i \neq j$ and $y_k \neq l$. (For the derivation, see [24].)

Given a new test data point \mathbf{x} , we have the multiple outputs $g_{\text{new}}^{y_{\text{new}}, l}$ for $y_{\text{new}} \neq l$. For the simplicity we try to get the estimates of $g_{\text{new}}^{1, l}$ ($l \neq 1$). For mutual k -NN classification, we have the predictive mean as in typical GP regression, as follows.

$$\mu_{g_{\text{new}, \text{MkNN}}^{1, l}} = \mu_{f_{\text{new}, \text{MkNN}}^1} - \mu_{f_{\text{new}, \text{MkNN}}^l} \quad (58)$$

$$= -\frac{1}{\kappa_{l, \text{MkNN}}} \mathbf{k}_{l, \text{MkNN}}^T \mathbf{1}, \quad (59)$$

where $\kappa_{l, \text{MkNN}}$ and $\mathbf{k}_{l, \text{MkNN}}$ are obtained from \mathbf{C}^{MUL} with the function w_{MkNN} and $\mathbf{C}_{J(n-1) \times J(n-1)+1}^{\text{MUL}}$ with one additional $g_{\text{new}, \text{MkNN}}^{1, l}$ [18], [19]. Based on $\{\mu_{g_{\text{new}, \text{MkNN}}^{1, l}} | l \neq 1\}$, we have the classification method based on the result of multivariate Bayesian MkNN regression.

$$y_{\text{new}}^{\text{MkNN, MUL-II}} = \begin{cases} C_1 & \text{if } \mu_{g_{\text{new}, \text{MkNN}}^{1, l}} > 0 \\ & \text{for all } l \neq 1; \\ C_{\arg\min_l \mu_{g_{\text{new}, \text{MkNN}}^{1, l}}} & \text{otherwise} \end{cases} \quad (60)$$

$$= C_{\arg\max_l \mu_{f_{\text{new}, \text{MkNN}}^l}} \quad (61)$$

$$= y_{\text{new}}^{\text{MkNN, MUL-I}} \quad (62)$$

Similarly, for symmetric k -NN classification we have the predictive mean as in typical GP regression, as follows.

$$\mu_{g_{\text{new}, \text{SkNN}}^{1, l}} = \mu_{f_{\text{new}, \text{SkNN}}^1} - \mu_{f_{\text{new}, \text{SkNN}}^l} \quad (63)$$

$$= -\frac{1}{\kappa_{l, \text{SkNN}}} \mathbf{k}_{l, \text{SkNN}}^T \mathbf{1}, \quad (64)$$

where $\kappa_{l, \text{SkNN}}$ and $\mathbf{k}_{l, \text{SkNN}}$ are obtained from \mathbf{C}^{MUL} with the function w_{SkNN} and $\mathbf{C}_{J(n-1) \times J(n-1)+1}^{\text{MUL}}$ with one additional $g_{\text{new}, \text{SkNN}}^{1, l}$ [18], [19]. Based on $\{\mu_{g_{\text{new}, \text{SkNN}}^{1, l}} | l \neq 1\}$, we have the classification method based on the result of multivariate Bayesian SkNN regression.

$$y_{\text{new}}^{\text{SkNN, MUL-II}} = \begin{cases} C_1 & \text{if } \mu_{g_{\text{new}, \text{SkNN}}^{1, l}} > 0 \\ & \text{for all } l \neq 1; \\ C_{\arg\min_l \mu_{g_{\text{new}, \text{SkNN}}^{1, l}}} & \text{otherwise} \end{cases} \quad (65)$$

$$= C_{\arg\max_l \mu_{f_{\text{new}, \text{SkNN}}^l}} \quad (66)$$

$$= y_{\text{new}}^{\text{SkNN, MUL-I}} \quad (67)$$

This latter formulation (*formulation II*) exactly leads to the binary classification formulation described in Section III-B1, when J is 2.

As can be seen in Eq (62) and Eq (67), both the formulations produce the same classification results when they have the same hyperparameters. The hyperparameters including k

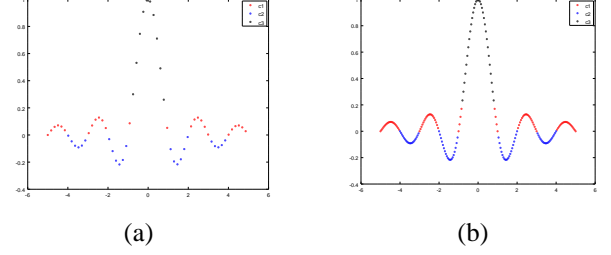


Fig. 1. The plots of the Sinc3C data set: (a) the training set, (b) the test set

can be selected by the methods described in Section II-B4. However, the hyperparameters selected by the methods with the *formulation I and II* can be different because they use different covariance matrixes in the marginal likelihood. (The former one uses the $Jn \times Jn$ covariance matrix and the latter one uses $(J - 1)n \times (J - 1)n$ covariance matrix.)

In the computer simulations even with the identical k there can be cases where the classification results by MkNN (or SkNN), the ones based on Bayesian MkNN (or SkNN) regression methods with *formulation I*, and the ones by Bayesian MkNN (or SkNN) regression methods with *formulation II* are different. One of the reasons for that is that the matrix calculation is approximate. Another reason is that they are different in the ways how dealing with vote tie cases. When vote ties occur, in MkNN (or SkNN) the class label of the nearest neighbor among tied mutual neighbors (or among tied symmetric neighbors) is assigned. However, in the methods based on Bayesian MkNN (or SkNN) regression, the class label with the lowest index is assigned, because information on nearest neighbors are not available in themselves.

IV. SIMULATION RESULTS

To demonstrate the proposed methods first we did simulations for an artificial data set. To generate an artificial data set, we used the equation $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ for the sinc function. We took the points equally spaced with the interval 0.17 between -5 and 5. We assigned class labels 1, 2, 3 to those points according to intervals which the function values at those points belong to among $(-\infty, 0)$, $[0, 0.2)$, $[0.2, \infty)$. We made up the training set with those points as inputs and with the assigned labels as target values. The data set is plotted in Figure 1. We call this data set the Sinc3C data set.

We applied MkNN and SkNN classification methods based on Bayesian MkNN and SkNN regression methods. We used both the *formulation I* requiring J outputs and the *formulation II* requiring $J - 1$ outputs. We tried the simulation repeatedly with different initial values for σ_0, σ , and found that one of the lowest marginal likelihoods is reached with the initial value 300, 3. We also applied MkNN and SkNN classification methods with k selected in the proposed methods, respectively, for each formulation. For comparison, we also applied k -NN³, MkNN , SkNN classification methods with the parameter k selected by the leave-one-out cross-validation method.

³When vote ties occur, in k -NN classification the class label of the nearest neighbor among tied neighbors is assigned.

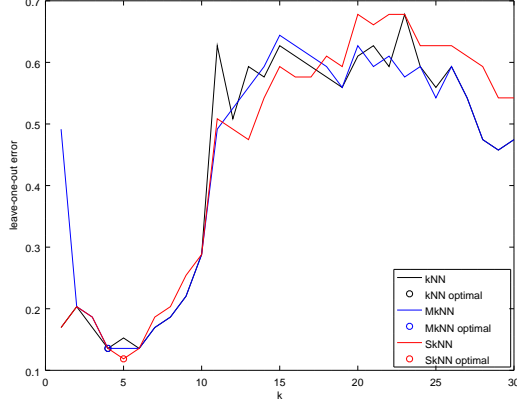


Fig. 2. The leave-one-out errors of k -NN, mutual k -NN, symmetric k -NN classification according to k for the Sinc3C training set. The points 'o' represent optimal ones.

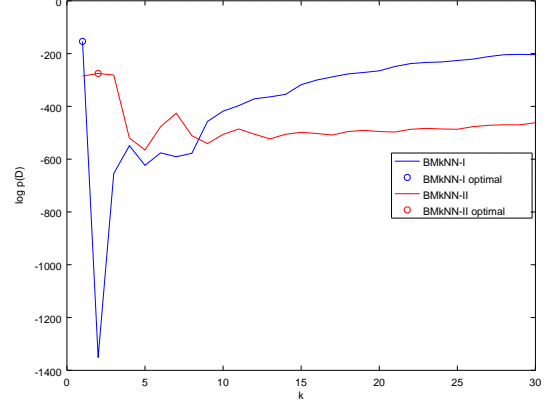
Figure 2 shows the leave-out-errors of k -NN, $MkNN$, $SkNN$ classification methods for the Sinc3C training set according to the parameter k . Figure 3 shows the log evidence of $BMkNN$, $BSkNN$ regression models with the multi-class formulation I and II for the Sinc3C training set according to the parameter k . $BMkNN$ -I and $BSkNN$ -I represent $MkNN$ and $SkNN$ classification with the formulation I based on Bayesian $MkNN$ and $SkNN$ regression, respectively. Likewise, $BMkNN$ -II and $BSkNN$ -II represent $MkNN$ and $SkNN$ classification with the formulation II based on Bayesian $MkNN$ and $SkNN$ regression, respectively.

Table I shows the classification error rates and k selected for all the methods applied to the Sinc3C data set. $MkNN$ (B-I k) and $SkNN$ (B-I k) represent $MkNN$ and $SkNN$ classification with the parameter k selected in $BMkNN$ -I, and $BSkNN$ -I, respectively. $MkNN$ (B-II k) and $SkNN$ (B-II k) represent $MkNN$ and $SkNN$ classification with the parameter k selected in $BMkNN$ -II, and $BSkNN$ -II, respectively. As can be seen in Table I, $BMkNN$ -II, $BSkNN$ -II, $MkNN$ (B-I k), and $SkNN$ (B-II k) perform significantly better than all the other methods.

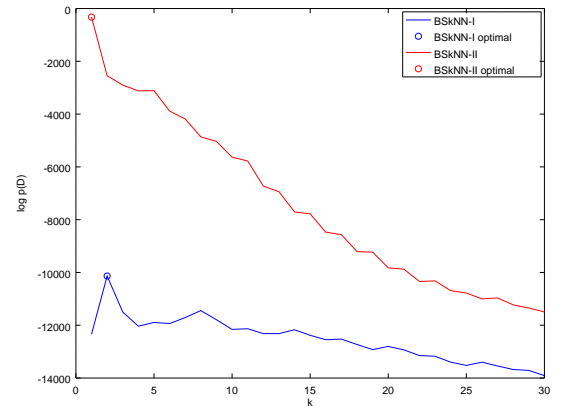
TABLE I
CLASSIFICATION ERROR RATES AND THE PARAMETER k 'S BY VARIOUS METHODS FOR THE SINC3C DATA SET

Methods	MSE	k selected
k -NN	0.079602	2
$MkNN$	0.064677	3
$SkNN$	0.064677	5
$BMkNN$ -I	0.059701	1
$BSkNN$ -I	0.089552	2
$MkNN$ (B-I k)	0.029851	1
$SkNN$ (B-I k)	0.074627	2
$BMkNN$ -II	0.029851	2
$BSkNN$ -II	0.029851	1
$MkNN$ (B-II k)	0.074627	2
$SkNN$ (B-II k)	0.029851	1

We applied the proposed methods and all the other methods to two real world data sets. As the first real world data set,



(a)



(b)

Fig. 3. (a) k vs. log evidence for $BMkNN$ -I and $BMkNN$ -II, (b) k vs. log evidence for $BSkNN$ -I and $BSkNN$ -II : The points 'o' represent the optimal ones

we use the Pima data set⁴. We used only the training set. It has 200 instances, 7 real-valued attributes, and 2 classes. We did 10 fold cross-validation to evaluate the performances of all the methods applied for the data set. The best results were obtained when the initial values of (σ_0, σ^2) were set to $(1, 10^{-6})$ for $BSkNN$, and when (σ_0, σ^2) was fixed to $(150, 1.5)$ for $BMkNN$.

Table II shows the parameter k 's selected by various methods such as cross-validation for k -NN, $MkNN$, $SkNN$ classifications, and the proposed methods (binary-class case) for $MkNN$ and $SkNN$ classifications. The abbreviation for the methods is the same as in Table I except that this case has only the one (binary-class) formulation rather than two formulations. Table III shows the means and standard deviations of mean squared errors (for 10 fold cross-validation) for the Pima data set. As can be seen in Table III, $MkNN$ and $BMkNN$ perform better than all the other methods.

To show how the methods work for the real world data set with more than two classes, we use New Thyroid data set [25]. It has 215 instances, 5 real-valued attributes, and 3 classes. We did 10 fold cross-validation to evaluate the performances

⁴Available from <https://www.stats.ox.ac.uk/pub/PRNN/>

TABLE II

THE PARAMETER k 'S SELECTED FOR EACH FOLD OF THE PIMA DATA SET BY VARIOUS METHODS: CROSS-VALIDATION FOR k -NN, $MkNN$, $SkNN$ CLASSIFICATIONS, AND THE PROPOSED METHODS (BINARY-CLASS CASES) FOR $MkNN$ AND $SkNN$ CLASSIFICATIONS

Methods	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	Mean
k -NN (CV)	5	14	12	12	10	5	12	14	16	6	10.6
$MkNN$ (CV)	49	27	75	73	78	80	79	70	84	69	68.4
$SkNN$ (CV)	51	48	47	53	5	36	36	84	11	53	42.4
$BMkNN$ (binary)	93	108	97	101	100	98	94	96	83	97	96.7
$BSkNN$ (binary)	42	35	33	33	31	32	36	33	38	34	34.7

TABLE III

MEANS AND STANDARD DEVIATIONS OF CLASSIFICATION ERROR RATES OF VARIOUS METHODS FOR THE PIMA DATA SET

Methods	MSE ($\mu \pm \sigma$)
k -NN	0.30500 \pm 0.064334
$MkNN$	0.25000 \pm 0.070711
$SkNN$	0.26000 \pm 0.077460
$BMkNN$ (binary-class)	0.25000 \pm 0.074536
$BSkNN$ (binary-class)	0.25500 \pm 0.072457
$MkNN$ (B k)	0.25500 \pm 0.089598
$SkNN$ (B k)	0.25500 \pm 0.072457

of all the methods applied for the data set. The best results were obtained when the initial values of (σ_0, σ^2) were set to $(100, 1)$, $(1, 0.01)$, and $(100, 1)$ for $BSkNN$ -I, $BMkNN$ -II, and $BSkNN$ -II, respectively. In case of $BMkNN$ -I (σ_0, σ^2) were set and fixed to $(1, 0.0001)$.

Table IV shows the parameter k 's selected by various methods such as cross-validation for k -NN, $MkNN$, $SkNN$ classifications, and the proposed methods (formulation I and II) for $MkNN$ and $SkNN$ classifications. The abbreviation for the methods is the same as in Table I. Table V shows the means and standard deviations of mean squared errors (for 10 fold cross-validation) for the New Thyroid data set. As can be seen in Table V, $BSkNN$ -I perform better than all the other methods.

TABLE IV

THE PARAMETER k 'S SELECTED FOR EACH FOLD OF THE NEW THYROID DATA SET BY VARIOUS METHODS: CROSS-VALIDATION FOR k -NN, $MkNN$, $SkNN$ CLASSIFICATIONS, AND THE PROPOSED METHODS (FORMULATION I AND II) FOR $MkNN$ AND $SkNN$ CLASSIFICATIONS

Methods	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	Mean
k -NN (CV)	1	4	1	1	1	1	2	1	1	1	1.4
$MkNN$ (CV)	42	39	43	27	25	4	60	24	43	35	34.2
$SkNN$ (CV)	1	5	2	2	2	2	2	2	2	2	2.2
$BMkNN$ -I	22	22	24	19	22	18	19	19	22	22	20.9
$BSkNN$ -I	4	2	5	4	1	2	4	2	4	4	3.2
$BMkNN$ -II	10	10	8	10	10	11	10	10	10	10	9.9
$BSkNN$ -II	1	2	2	2	2	2	2	2	2	2	1.9

V. CONCLUSION

We have proposed symmetric k -NN classification method, which is another variate of k -NN classification method. We have proposed methods to select the parameter k in mutual and symmetric k -NN classification methods. The selection problems boil down to the ones for the parameter k in Bayesian mutual and symmetric k -NN regression methods, because Bayesian mutual and symmetric k -NN classifications can be done by Bayesian mutual and symmetric k -NN regression methods with new multiple-output encodings of target values.

TABLE V

MEANS AND STANDARD DEVIATIONS OF CLASSIFICATION ERROR RATES OF VARIOUS METHODS FOR THE NEW THYROID DATA SET

Methods	MSE ($\mu \pm \sigma$)
k -NN	0.041991 \pm 0.040510
$MkNN$	0.050866 \pm 0.062227
$SkNN$	0.046537 \pm 0.053345
$BMkNN$ -I	0.046320 \pm 0.060985
$BSkNN$ -I	0.032684 \pm 0.031835
$MkNN$ (B-I k)	0.050866 \pm 0.066152
$SkNN$ (B-I k)	0.037229 \pm 0.047617
$BMkNN$ -II	0.037229 \pm 0.053066
$BSkNN$ -II	0.042208 \pm 0.046817
$MkNN$ (B-II k)	0.046320 \pm 0.065328
$SkNN$ (B-II k)	0.037446 \pm 0.043098

For that purpose two kinds of encodings were proposed. The simulation results showed the proposed methods is comparable to or better than the selection by the leave-one-out cross-validation methods.

APPENDIX A

PROOF OF THEOREM 3

Theorem 3 can be done similarly to the proof of Theorem 1 in [22] as follows.

(1) Since Laplacian matrix $\mathbf{L} (= \mathbf{D} - \mathbf{W})$ is positive semidefinite [26], for $\sigma^2 > 0$ $\tilde{\mathbf{C}} (= \mathbf{L} + \sigma^2 \mathbf{I})$ is positive definite. So $\tilde{\mathbf{C}}$ is positive definite.

(2) Since $\tilde{\mathbf{C}}^T = (\mathbf{D} - \mathbf{W} + \sigma^2 \mathbf{I})^T = \mathbf{D}^T - \mathbf{W}^T + \sigma^2 \mathbf{I}^T = \mathbf{D} - \mathbf{W} + \sigma^2 \mathbf{I} = \tilde{\mathbf{C}}$, $\tilde{\mathbf{C}}$ is symmetric.

From (1) & (2), by Theorem 7.5 in [27] $\tilde{\mathbf{C}}$ is a valid covariance matrix. QED.

APPENDIX B

APPENDIX B. PROOF OF THEOREM 4

In case $\sum_{i=1}^N \{\delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j}\} = 0$, it is trivial by Eq (3) and (23).

Otherwise, take a small positive $\epsilon \in m_n^{SkNNR}(\mathbf{x})$.

Set $\delta = [\sum_{i=1}^N \{\delta_{\mathbf{x}_j \sim_k \mathbf{x}_i} + \delta_{\mathbf{x}_i \sim_k \mathbf{x}_j}\}] / \{\frac{m_n^{SkNNR}(\mathbf{x})}{\epsilon} - 1\}$. Then, if $\|\sigma^2 / \sigma_0\| < \delta$, $\|\mu_{f_U, SkNN} - m_n^{SkNNR}(\mathbf{x})\| < \epsilon$. By the (ϵ, δ) definition of the limit of a function, we get the statement in the theorem. QED.

REFERENCES

- [1] E. Fix and J. L. Hodges Jr, "Discriminatory analysis-nonparametric discrimination: consistency properties," DTIC Document, Tech. Rep., 1951.
- [2] E. Fix and J. Hodges, "Discriminatory analysis: small sample performance," 1952.
- [3] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [4] R. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 622–627, 1981.
- [5] C. C. Holmes and N. M. Adams, "A probabilistic nearest neighbour method for statistical pattern recognition," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 64, no. 2, pp. 295–306, 2002.
- [6] A. K. Ghosh, "On optimum choice of k in nearest neighbor classification," *Computational Statistics & Data Analysis*, vol. 50, no. 11, pp. 3113–3123, 2006.
- [7] L. Cucala, J.-M. Marin, C. P. Robert, and D. M. Titterton, "A Bayesian Reassessment of Nearest-Neighbor Classification," *Journal of the American Statistical Association*, vol. 104, no. 485, pp. 263–273, 2009.
- [8] C. C. Holmes and N. M. Adams, "Likelihood inference in nearest-neighbour classification models," *Biometrika*, vol. 90, no. 1, pp. 99–112, 2003.
- [9] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern recognition*, vol. 10, no. 2, pp. 105–112, 1978.
- [10] —, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood," *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488–490, 1979.
- [11] H. Liu, S. Zhang, J. Zhao, X. Zhao, and Y. Mo, "A new classification algorithm using mutual nearest neighbors," in *9th International Conference on Grid and Cooperative Computing (GCC)*. IEEE, 2010, pp. 52–57.
- [12] V. Hautamäki, I. Kärkkäinen, and P. Fränti, "Outlier detection using k-nearest neighbour graph," in *ICPR (3)*, 2004, pp. 430–433.
- [13] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 2–11, 2010.
- [14] D. Guru and H. Nagendraswamy, "Clustering of interval-valued symbolic patterns based on mutual similarity value and the concept of k-mutual nearest neighborhood," in *Computer Vision-ACCV 2006*. Springer, 2006, pp. 234–243.
- [15] A. Guyader and N. Hengartner, "On the mutual nearest neighbors estimate in regression," *Journal of Machine Learning Research*, vol. 14, pp. 2361–2376, 2013.
- [16] K. Ozaki, M. Shimbo, M. Komachi, and Y. Matsumoto, "Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data," in *Proceedings of the fifteenth conference on computational natural language learning*. Association for Computational Linguistics, 2011, pp. 154–162.
- [17] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems*, vol. 8, 1995.
- [18] M. Gibbs and D. J. MacKay, "Efficient implementation of gaussian processes," Tech. Rep., 1997.
- [19] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.
- [20] R. Neal, "Regression and classification using gaussian process priors," *Bayesian Statistics*, vol. 6, pp. 475–501, 1997.
- [21] X. Zhu, J. D. Lafferty, and Z. Ghahramani, "Semi-supervised learning: From Gaussian fields to Gaussian processes," 2003.
- [22] H.-C. Kim, "Bayesian Kernel and Mutual k-Nearest Neighbor Regression," *ArXiv e-prints*, Aug. 2016.
- [23] C. K. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1342–1351, 1998.
- [24] H.-C. Kim and Z. Ghahramani, "Bayesian Gaussian Process Classification with the EM-EP Algorithm," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1948–1959, 2006.
- [25] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [26] R. Merris, "Laplacian matrices of graphs: a survey," *Linear algebra and its applications*, vol. 197, pp. 143–176, 1994.
- [27] D. Stefanica, *A Linear Algebra Primer for Financial Engineering*. Fe Press, 2014.